

### AMENDMENTS TO THE CLAIMS

1. (Currently amended) An article comprising a machine-accessible medium having stored thereon instructions that, when executed by a machine, cause the machine to:
  - execute a host code in a host operating system environment;
  - create a plurality of virtual machines in a virtual environment, the virtual environment being a direct execution environment;
  - transfer a virtual code from the host operating system environment to the virtual environment;
  - execute virtual code on at least one of the plurality of virtual machines; and
  - provide a single monitor within the host operating system environment to control entry to and exit from each of the plurality of virtual machines in the direct execution environment.
2. (Canceled)
3. (Canceled)
4. (Currently amended) The article of claim 1, having further instructions that, when executed by the machine, cause the monitor to:
  - control transfer of virtual code between the host operating system environment and the virtual environment based on a virtualization event attempted by at least one of the virtual machines.
5. (Original) The article of claim 4, having further instructions that, when executed by the machine, cause the monitor to gain control over the virtualization event from the direct execution environment.
6. (Original) The article of claim 5, having further instructions that, when executed by the machine, cause the monitor to return execution to the direct execution environment after a virtualization operation.
7. (Currently amended) The article of claim 5, having further instructions that, when executed by the machine, cause the monitor to pass control to a platform simulator within the host operating system environment for simulation of the virtualization event.
8. (Original) The article of claim 4, having further instructions that, when executed by the machine, cause the monitor to access a list of virtualization events.

9. (Previously presented) The article of claim 1, having further instructions that, when executed by the machine, cause the monitor to:

in response to an exit from the direct execution environment, store state data; and  
restore the stored state data prior to entry to the direct execution environment.

10. (Original) The article of claim 1, wherein the virtual code includes a plurality of virtual codes each executing on a separate one of the plurality of virtual machines.

11. (Currently amended) A method comprising:

accessing simulated instruction codes in a host operating system environment operating on a central processing unit (CPU) implementing Virtual Machine Extensions;

launching a plurality of virtual machines in a virtual environment on the CPU, the virtual environment being a direct execution environment;

virtualizing a CPU state associated with the simulated instruction codes;

executing at least one of the simulated instruction codes on at least one of the plurality of virtual machines; and

monitoring, via a single monitor, each of the plurality of virtual machines from within the host operating system environment to control entry to and exit from each of the plurality of virtual machines.

12. (Original) The method of claim 11 further comprising:

detecting an occurrence of a virtualization event in any one of the plurality of virtual machines;

in response to detecting the virtualization event, exiting the virtual environment; and  
analyzing the virtualization event.

13. (Original) The method of claim 12 further comprising:

determining whether the virtualization event is a complex event; and

if the virtualization event is not a complex event, virtualizing the simulated instruction code associated with the virtualization event.

14. (Original) The method of claim 13 further comprising re-entering the virtual environment after the simulated instruction code associated with the virtualization event is virtualized.

15. (Original) The method of claim 13 further comprising:

if the virtualization event is a complex event, de-virtualizing the CPU state; and  
simulating the simulated instruction code associated with the virtualization event.

16. (Original) The method of claim 12, further comprising:

storing the CPU state upon exiting the virtual environment; and  
restoring the stored CPU state upon re-entering the virtual environment.

17. (Currently amended) A system comprising:

hardware to generate and control a plurality of virtual machines that each are capable of executing simulated instruction code, wherein the hardware is able to create an abstraction of a real machine for executing a real operating system on the computer system;

a direct execution environment to execute the simulated instruction codes and associated data as virtual codes;

a plurality of virtual machines formed within the direct execution environment; and

a host operating system environment comprising a single monitor for controlling exit from and entry to the plurality of virtual machines formed within the direct execution environment.

18. (Currently amended) The system of claim 17, wherein the monitor generates the plurality of virtual machines and performs virtualization operations, and wherein the host operating system environment comprises a platform simulator to perform simulations of virtualization events.

19. (Previously presented) The system of claim 18, wherein the monitor gains control from the direct execution environment whenever at least one of the plurality of virtual machines attempts to perform a virtualization event, and wherein the monitor switches from one of the plurality of virtual machines to another of the plurality of virtual machines in response to an analysis of the virtualization event.

20. (Previously presented) The article of claim 1, having further instructions that, when executed by the machine, cause the monitor to:

assign the virtual code to any one of the plurality of virtual machines for execution; and  
monitor the plurality of virtual machines for a virtualization event.

21. (Currently amended) The article of claim 20, having further instructions that, when executed by the machine, cause the monitor to:

determine if the virtualization event is an end of instructions quota event for one of the plurality of virtual machines; and

in response to an identification of the end of instructions quota event, switch to another one of the plurality of virtual machines.

22. (Currently amended) The method of claim 12, further comprising:

determining if the virtualization event is an end of instructions quota event for one of the plurality of virtual machines; and

in response to an identification of the end of instructions quota event, switching to another one of the plurality of virtual machines.